**Creating Microservices for account and loan**

package com.cognizant.account;

import org.springframework.web.bind.annotation.\*;

*@RestController*

*@RequestMapping*("/accounts")

public class AccountController {

*@GetMapping*("/{number}")

public Account getAccountDetails(*@PathVariable* String number) {

return new Account(number, "savings", 234343);

}

public static class Account {

private String number;

private String type;

private double balance;

public Account(String number, String type, double balance) {

this.number = number;

this.type = type;

this.balance = balance;

}

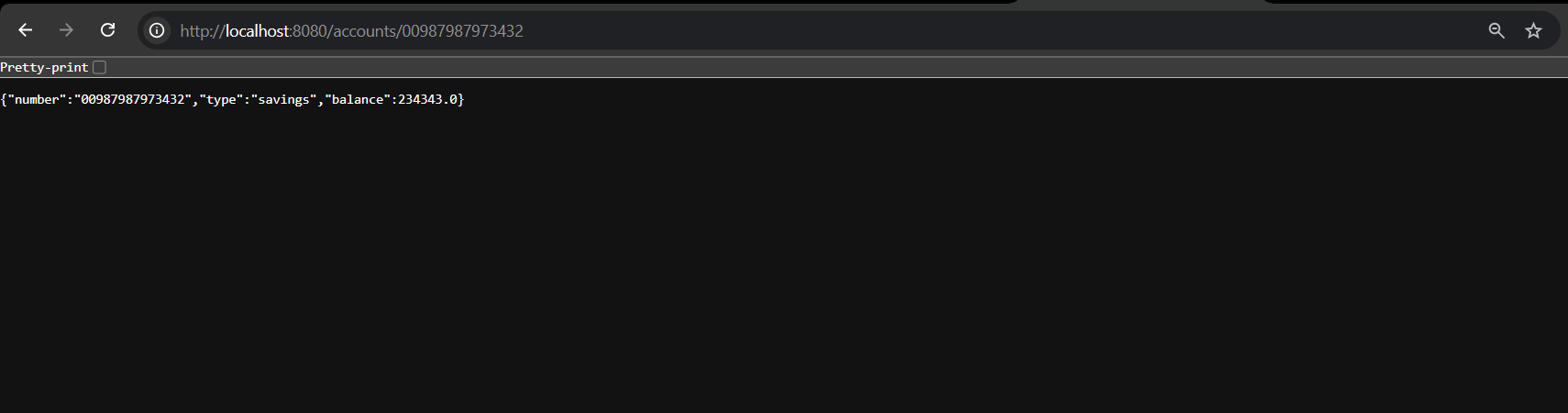
public String getNumber() { return number; }

public String getType() { return type; }

public double getBalance() { return balance; }

}

}



package com.cognizant.loan;

import org.springframework.web.bind.annotation.\*;

*@RestController*

*@RequestMapping*("/loans")

public class LoanController {

*@GetMapping*("/{number}")

public Loan getLoanDetails(*@PathVariable* String number) {

return new Loan(number, "car", 400000, 3258, 18);

}

public static class Loan {

private String number;

private String type;

private double loan;

private double emi;

private int tenure;

public Loan(String number, String type, double loan, double emi, int tenure) {

this.number = number;

this.type = type;

this.loan = loan;

this.emi = emi;

this.tenure = tenure;

}

public String getNumber() { return number; }

public String getType() { return type; }

public double getLoan() { return loan; }

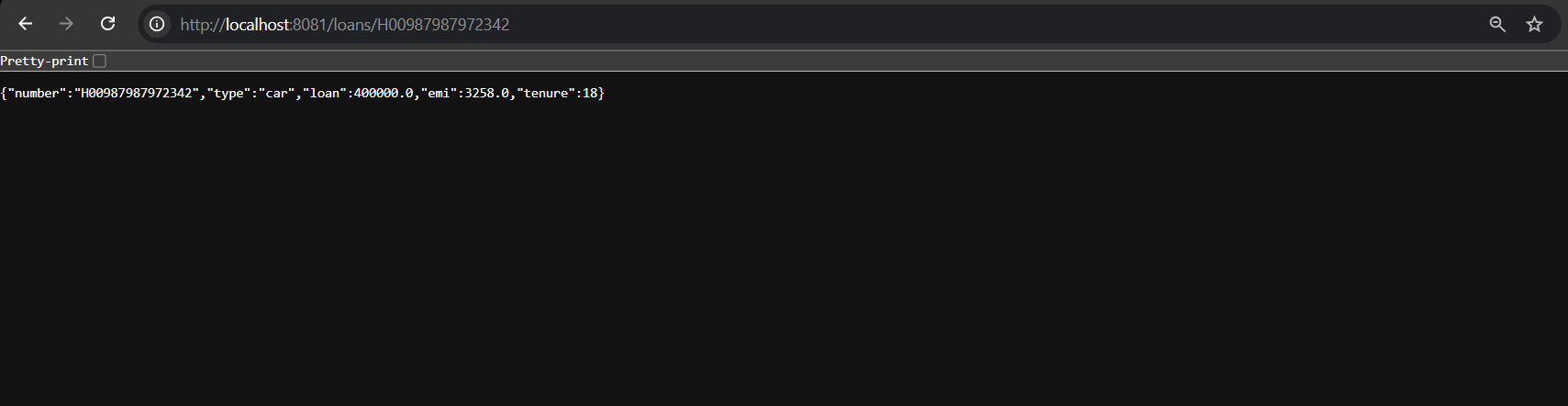
public double getEmi() { return emi; }

public int getTenure() { return tenure; }

}

}

server.port=8081



**Create Eureka Discovery Server and register microservices**

**package com.cognizant.eureka\_discovery\_server;**

**import org.springframework.boot.SpringApplication;**

**import org.springframework.boot.autoconfigure.SpringBootApplication;**

**import org.springframework.cloud.netflix.eureka.server.EnableEurekaServer;**

***@EnableEurekaServer***

***@SpringBootApplication***

**public class EurekaDiscoveryServerApplication {**

**public static void main(String[] args) {**

**SpringApplication.*run*(EurekaDiscoveryServerApplication.class, args);**

**}**

**}**

**server.port=8761**

**eureka.client.register-with-eureka=false**

**eureka.client.fetch-registry=false**

**logging.level.com.netflix.eureka=OFF**

**logging.level.com.netflix.discovery=OFF**

**spring.devtools.restart.enabled=false**

**package com.cognizant.account;**

**import org.springframework.boot.SpringApplication;**

**import org.springframework.boot.autoconfigure.SpringBootApplication;**

**import org.springframework.cloud.client.discovery.EnableDiscoveryClient;**

***@EnableDiscoveryClient***

***@SpringBootApplication***

**public class AccountApplication {**

**public static void main(String[] args) {**

**SpringApplication.*run*(AccountApplication.class, args);**

**}**

**}**

**package com.cognizant.loan;**

**import org.springframework.boot.SpringApplication;**

**import org.springframework.boot.autoconfigure.SpringBootApplication;**

**import org.springframework.cloud.client.discovery.EnableDiscoveryClient;**

***@EnableDiscoveryClient***

***@SpringBootApplication***

**public class LoanApplication {**

**public static void main(String[] args) {**

**SpringApplication.*run*(LoanApplication.class, args);**

**}**

**}**

**server.port=8080**

**spring.application.name=account-service**

**eureka.client.service-url.defaultZone=http://localhost:8761/eureka**

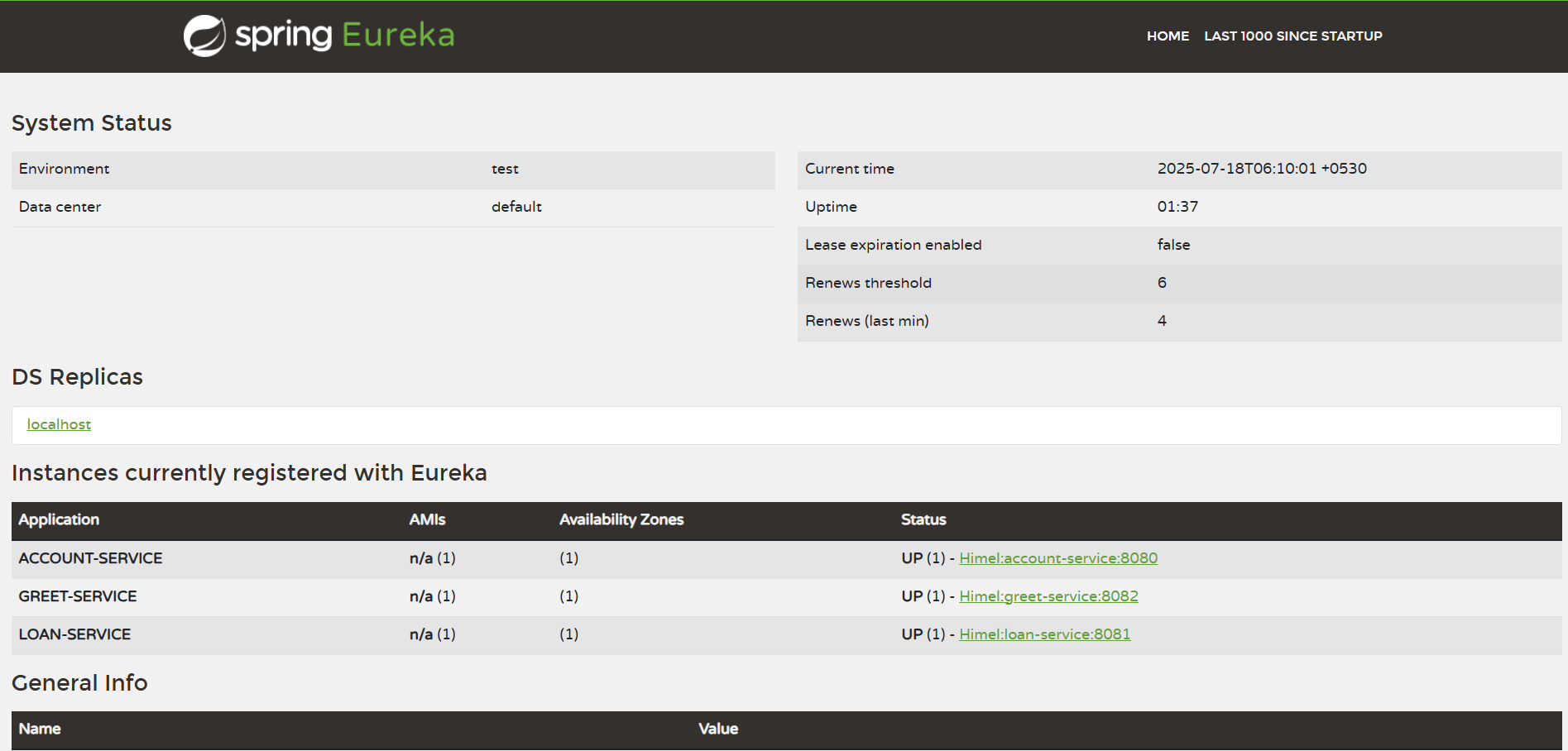
**spring.devtools.restart.enabled=false**

**server.port=8081**

**spring.application.name=loan-service**

**eureka.client.service-url.defaultZone=http://localhost:8761/eureka**

**spring.devtools.restart.enabled=false**

****

**server.port=9090**

**spring.application.name=api-gateway**

**eureka.client.service-url.defaultZone=http://localhost:8761/eureka**

**spring.cloud.gateway.discovery.locator.enabled=true**

**spring.cloud.gateway.discovery.locator.lowerCaseServiceId=true**

**spring.devtools.restart.enabled=false**

**package com.cognizant.api\_gateway;**

**import org.slf4j.Logger;**

**import org.slf4j.LoggerFactory;**

**import org.springframework.cloud.gateway.filter.GlobalFilter;**

**import org.springframework.core.Ordered;**

**import org.springframework.stereotype.Component;**

**import org.springframework.web.server.ServerWebExchange;**

**import reactor.core.publisher.Mono;**

***@Component***

**public class LogFilter implements GlobalFilter, Ordered {**

**private final Logger logger = LoggerFactory.*getLogger*(LogFilter.class);**

***@Override***

**public Mono<Void> filter(ServerWebExchange exchange, org.springframework.cloud.gateway.filter.GatewayFilterChain chain) {**

**logger.info("Incoming Request: " + exchange.getRequest().getURI());**

**return chain.filter(exchange);**

**}**

***@Override***

**public int getOrder() {**

**return -1;**

**}**

**}**